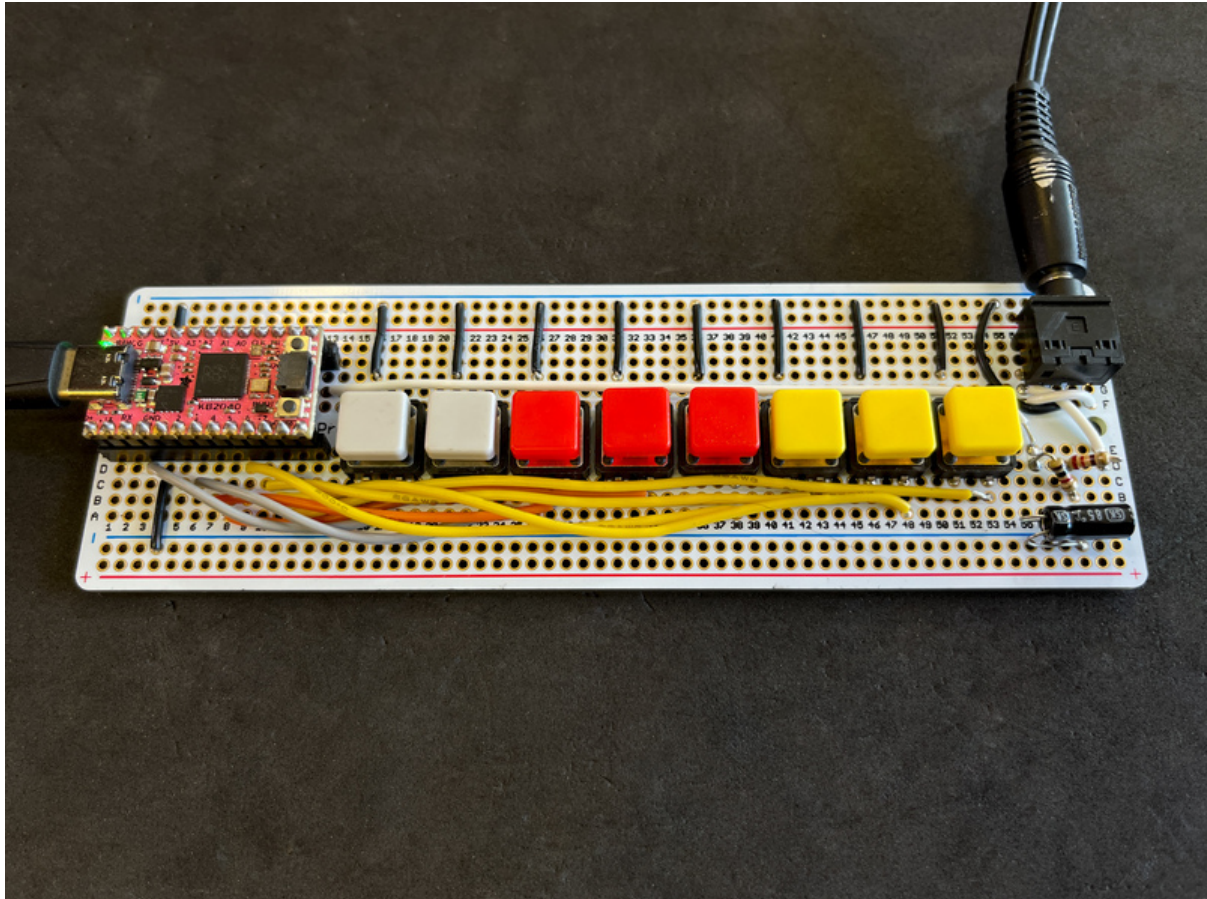




# Breakbeat Breadboard

Created by John Park



<https://learn.adafruit.com/breakbeat-breadboard>

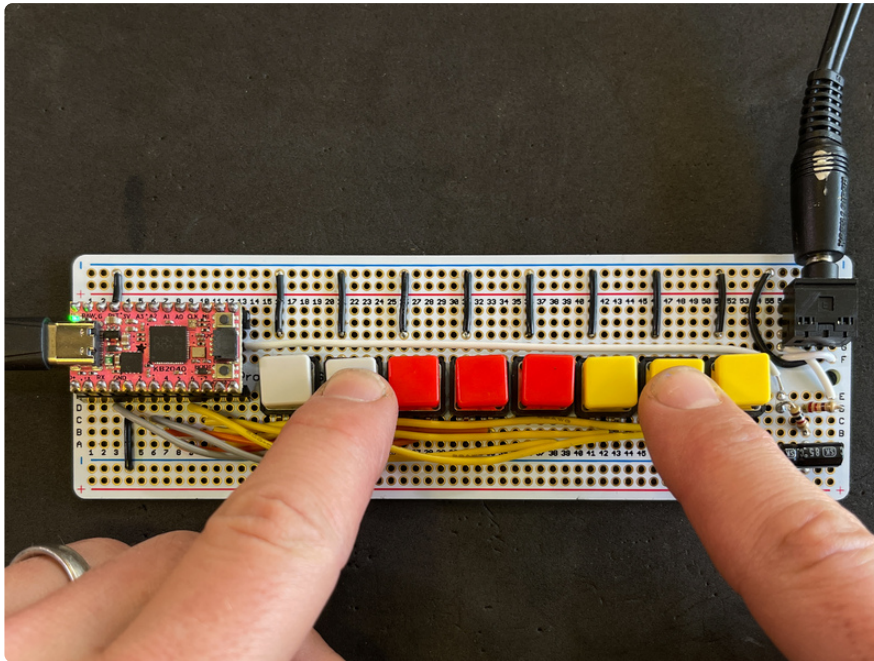
Last updated on 2024-06-03 03:35:05 PM EDT

# Table of Contents

Overview	3
<hr/>	
• Parts	
• Tools	
CircuitPython	6
<hr/>	
• CircuitPython Quickstart	
• Safe Mode	
• Flash Resetting UF2	
Build the Breakbeat Breadboard	10
<hr/>	
• Kee Boar Pins	
• Audio Noise Reduction	
• Buttons	
Code the Breakbeat Breadboard	15
<hr/>	
• Text Editor	
• Download the Project Bundle	
• Upload the Code and Libraries to the KB RP2040	
• Custom Sample Loops	

---

# Overview

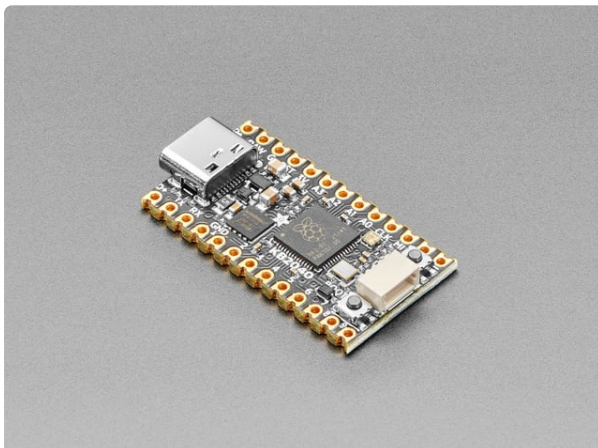


Remix breakbeats or other sample loops on the fly with the Breakbeat Breadboard! Thanks to the CircuitPython audio library mixer, it's easy to do!

This project is easy to build, and the colorful tactile buttons are irresistible to push. Hold any button or combination of buttons to un-mute the synced loops like a DJ killswitch -- the music loops play constantly in the background, the audio mixer takes care of the rest. No tricky sync code to worry about.

While this project uses eight buttons, the KB2040 has enough GPIO pins to comfortably increase this to sixteen buttons before requiring a diode matrix.

## Parts



### Adafruit KB2040 - RP2040 Kee Boar Driver

A wild Kee Boar appears! It's a shiny KB2040! An Arduino Pro Micro-shaped board for Keebs with RP2040. (#keeblife 4 evah) A lot of folks like using Adafruit... <https://www.adafruit.com/product/5302>

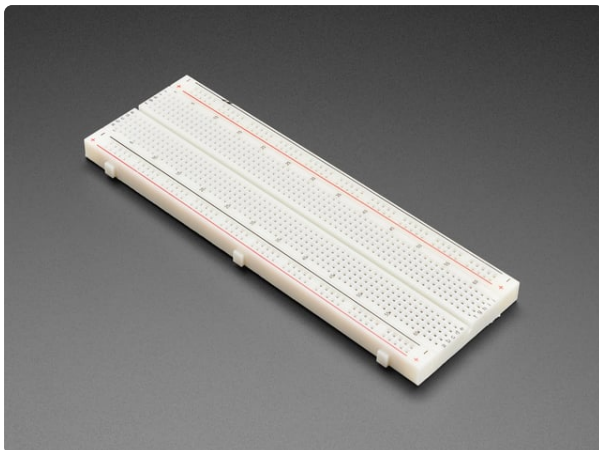




### Colorful 12mm Square Tactile Button Switch Assortment - 15 pack

Little clicky switches are standard input "buttons" on electronic projects. These work best in a PCB but can be...

<https://www.adafruit.com/product/1010>



### Full Sized Premium Breadboard - 830 Tie Points

This is a 'full-size' premium quality breadboard, 830 tie points. Good for small and medium projects. It's 2.2" x 7" (5.5 cm x 17 cm) with a standard double-strip...

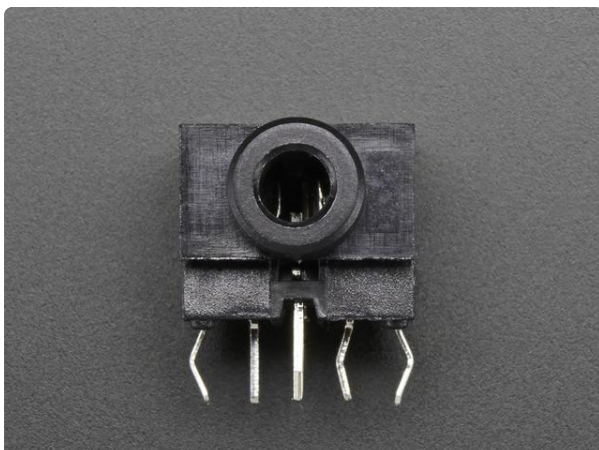
<https://www.adafruit.com/product/239>



### Adafruit Perma-Proto Full-sized Breadboard PCB - Single

Customers have asked us to carry basic perf-board, but we never liked the look of most basic perf: it's always crummy quality, with pads that flake off and no labeling. Then we...

<https://www.adafruit.com/product/1606>



### Breadboard-Friendly 3.5mm Stereo Headphone Jack

Pipe audio in or out of your project with this very handy breadboard-friendly audio jack. It's a stereo jack with disconnect-switches on Left and Right channels as well as a center...

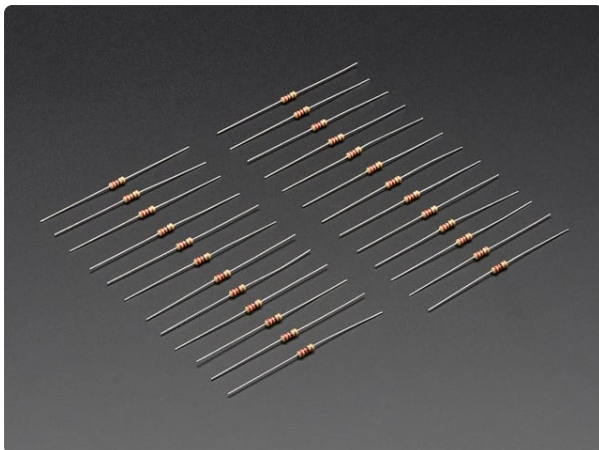
<https://www.adafruit.com/product/1699>



### Hook-up Wire Spool Set - 22AWG Solid Core - 6 x 25 ft

Perfect for bread-boarding, free wiring, etc. This box contains 6 spools of solid-core wire. The wire is easy to solder to and when bent it keeps its shape pretty well. We like to have...

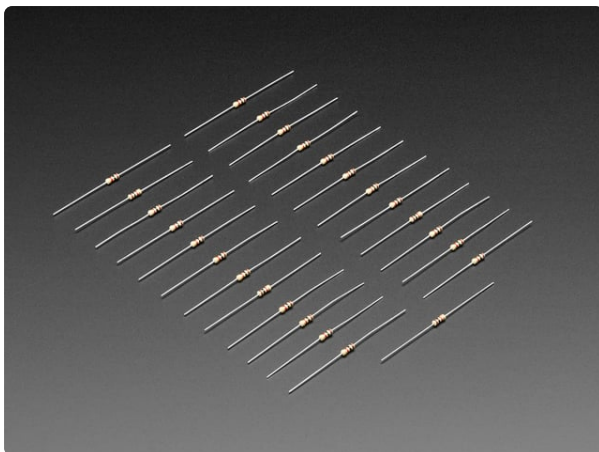
<https://www.adafruit.com/product/1311>



### Through-Hole Resistors - 220 ohm 5% 1/4W - Pack of 25

ΩMG! You're not going to be able to resist these handy resistor packs! Well, axially, they do all of the resisting for you! This is a 25 Pack of...

<https://www.adafruit.com/product/2780>



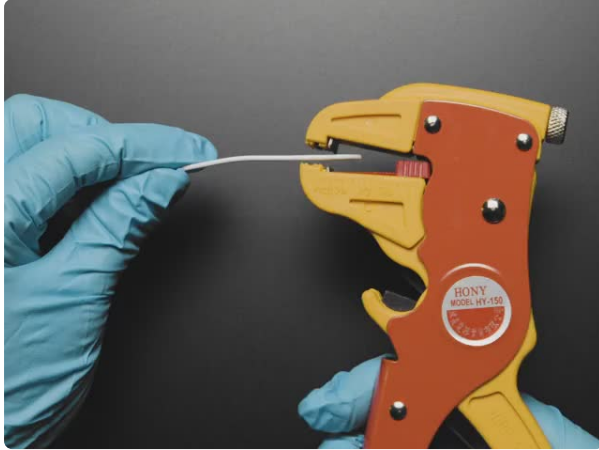
### Through-Hole Resistors - 1.0K ohm 5% 1/4W - Pack of 25

ΩMG! You're not going to be able to resist these handy resistor packs! Well, axially, they do all of the resisting for you! This is a 25 Pack of...

<https://www.adafruit.com/product/4294>

You'll also need a electrolytic through-hole capacitor for audio filtering, a 1 $\mu$ F or 2.2 $\mu$ F cap works well. This is just used to filter some of the high-end noise.

## Tools



### [Automatic Self-Adjusting Wire Strippers and Cutter](#)

If you've got a lot of wire-stripping to do, you may want to try out an automatic stripper/cutter! They are fantastic for stripping and cutting stranded and solid-core 24 or...

<https://www.adafruit.com/product/4747>

---

## CircuitPython

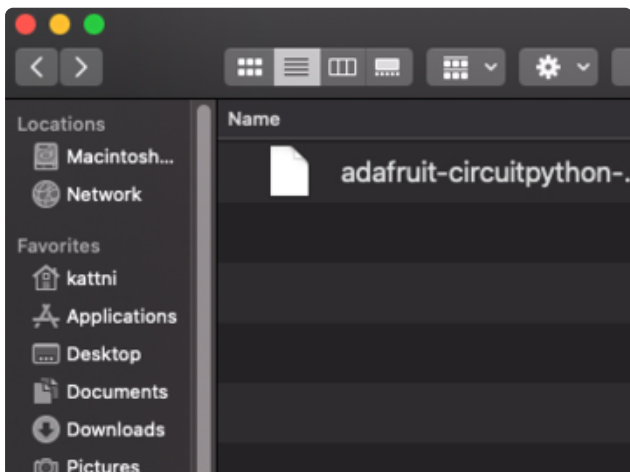
[CircuitPython](https://adafru.it/tB7) (<https://adafru.it/tB7>) is a derivative of [MicroPython](https://adafru.it/BeZ) (<https://adafru.it/BeZ>) designed to simplify experimentation and education on low-cost microcontrollers. It makes it easier than ever to get prototyping by requiring no upfront desktop software downloads. Simply copy and edit files on the **CIRCUITPY** drive to iterate.

## CircuitPython Quickstart

Follow this step-by-step to quickly get CircuitPython running on your board.

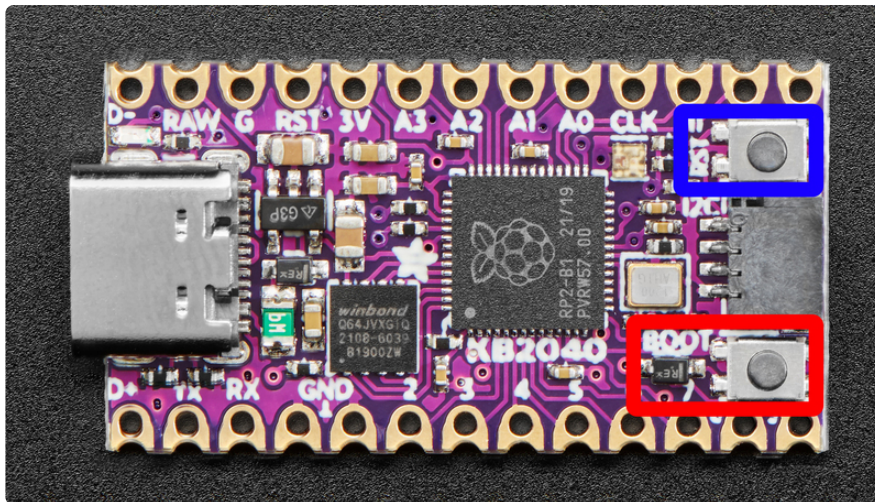
Download the latest version of  
CircuitPython for this board via  
[circuitpython.org](https://circuitpython.org)

<https://adafru.it/Xdr>



Click the link above to download the latest CircuitPython UF2 file.

Save it wherever is convenient for you.



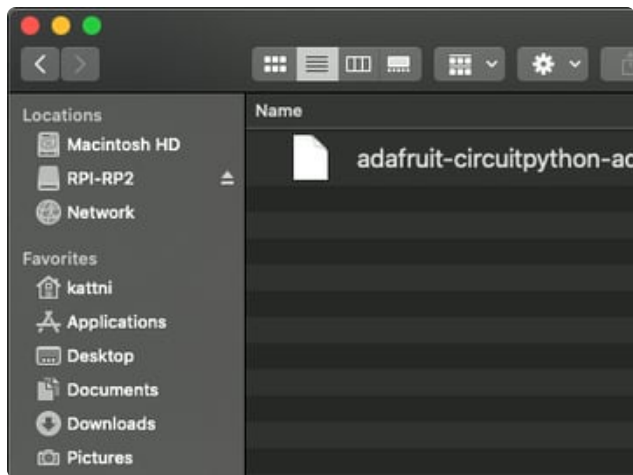
To enter the bootloader, hold down the **BOOT/BOOTSEL** button (highlighted in red above), and while continuing to hold it (don't let go!), press and release the **reset** button (highlighted in blue above). **Continue to hold the BOOT/BOOTSEL button until the RPI-RP2 drive appears!**

If the drive does not appear, release all the buttons, and then repeat the process above.

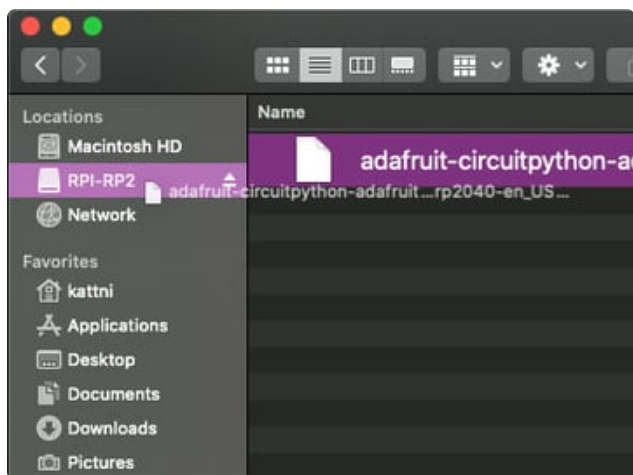
You can also start with your board unplugged from USB, press and hold the BOOTSEL button (highlighted in red above), continue to hold it while plugging it into USB, and wait for the drive to appear before releasing the button.

A lot of people end up using charge-only USB cables and it is very frustrating! **Make sure you have a USB cable you know is good for data sync.**

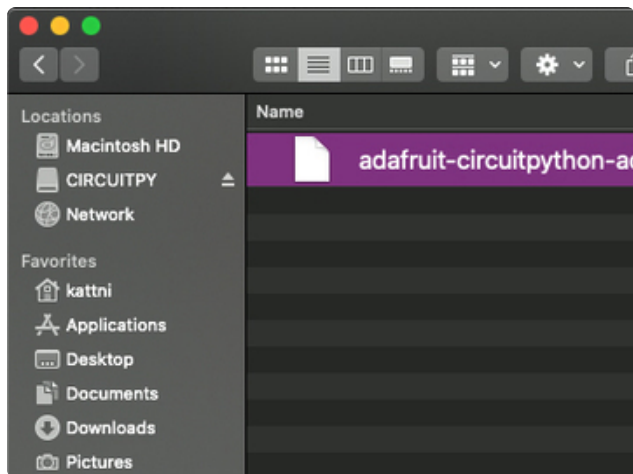




You will see a new disk drive appear called **RPI-RP2**.



Drag the **adafruit\_circuitpython\_etc.uf2** file to **RPI-RP2**.



The **RPI-RP2** drive will disappear and a new disk drive called **CIRCUITPY** will appear.

That's it, you're done! :)

## Safe Mode

You want to edit your **code.py** or modify the files on your **CIRCUITPY** drive, but find that you can't. Perhaps your board has gotten into a state where **CIRCUITPY** is read-only. You may have turned off the **CIRCUITPY** drive altogether. Whatever the reason, safe mode can help.



Safe mode in CircuitPython does not run any user code on startup, and disables auto-reload. This means a few things. First, safe mode bypasses any code in **boot.py** (where you can set **CIRCUITPY** read-only or turn it off completely). Second, it does not run the code in **code.py**. And finally, it does not automatically soft-reload when data is written to the **CIRCUITPY** drive.

Therefore, whatever you may have done to put your board in a non-interactive state, safe mode gives you the opportunity to correct it without losing all of the data on the **CIRCUITPY** drive.

## Entering Safe Mode

To enter safe mode when using CircuitPython, plug in your board or hit reset (highlighted in red above). Immediately after the board starts up or resets, it waits 1000ms. On some boards, the onboard status LED (highlighted in green above) will blink yellow during that time. If you press reset during that 1000ms, the board will start up in safe mode. It can be difficult to react to the yellow LED, so you may want to think of it simply as a slow double click of the reset button. (Remember, a fast double click of reset enters the bootloader.)

## In Safe Mode

If you successfully enter safe mode on CircuitPython, the LED will intermittently blink yellow three times.

If you connect to the serial console, you'll find the following message.

```
Auto-reload is off.  
Running in safe mode! Not running saved code.  
  
CircuitPython is in safe mode because you pressed the reset button during boot.  
Press again to exit safe mode.  
  
Press any key to enter the REPL. Use CTRL-D to reload.
```

You can now edit the contents of the **CIRCUITPY** drive. Remember, your code will not run until you press the reset button, or unplug and plug in your board, to get out of safe mode.

## Flash Resetting UF2

If your board ever gets into a really weird state and CIRCUITPY doesn't show up as a disk drive after installing CircuitPython, try loading this 'nuke' UF2 to RPI-RP2. which

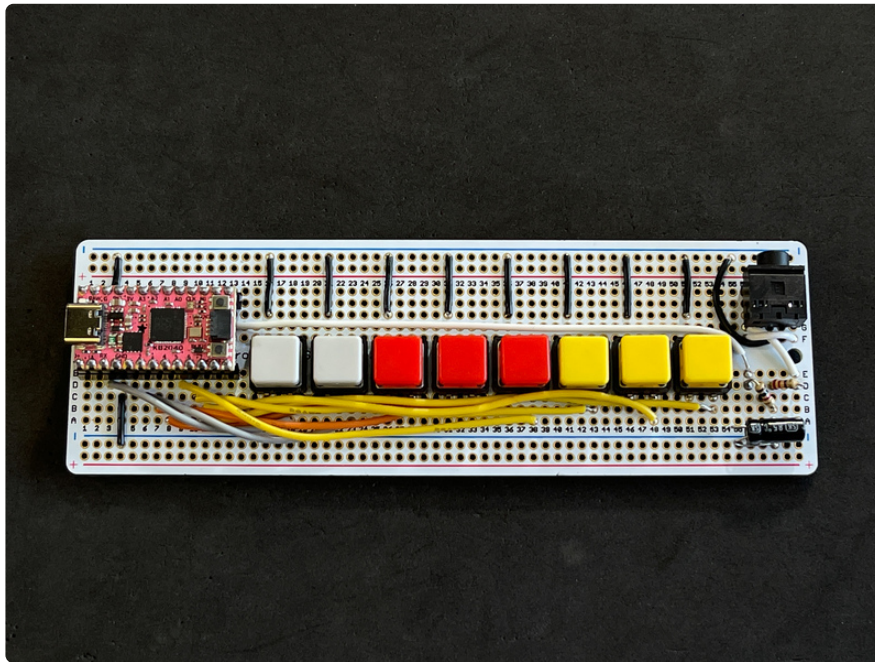
will do a 'deep clean' on your Flash Memory. **You will lose all the files on the board**, but at least you'll be able to revive it! After loading this UF2, follow the steps above to re-install CircuitPython.

Download flash erasing "nuke" UF2

<https://adafru.it/RLE>

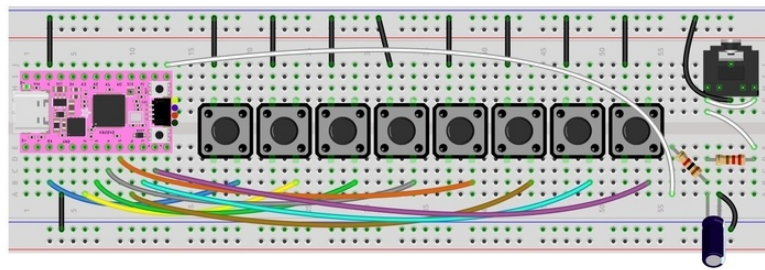
---

## Build the Breakbeat Breadboard

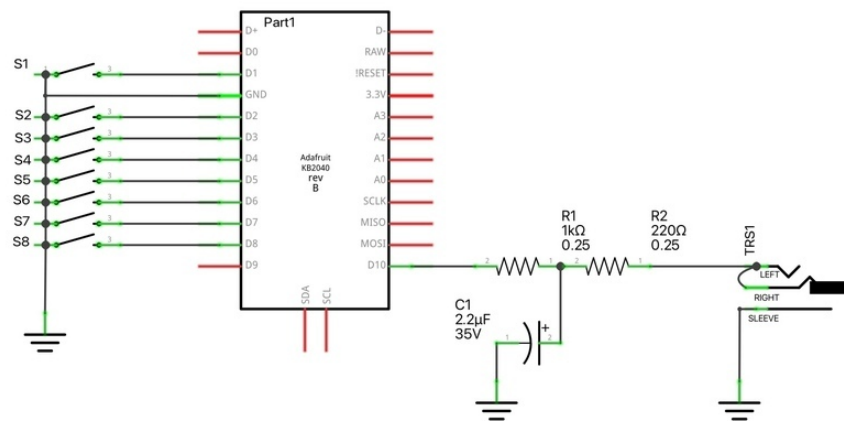


To build the Breakbeat Breadboard you can use either a breadboard or a Perma Proto board. The only real differences in the builds will be the use of optional header sockets for the Perma Proto build, and soldering wires to the Perma Proto instead of just poking wires into the breadboard.

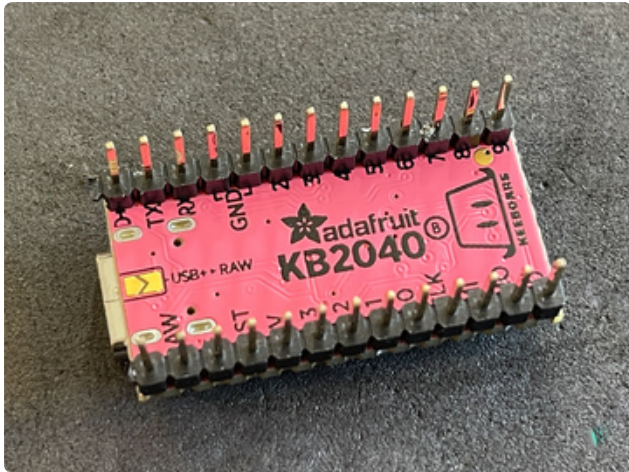
Follow the diagrams here for either build.



fritzing

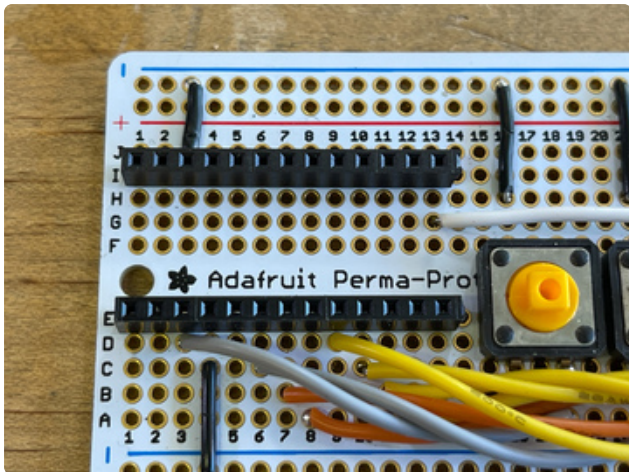


fritzing

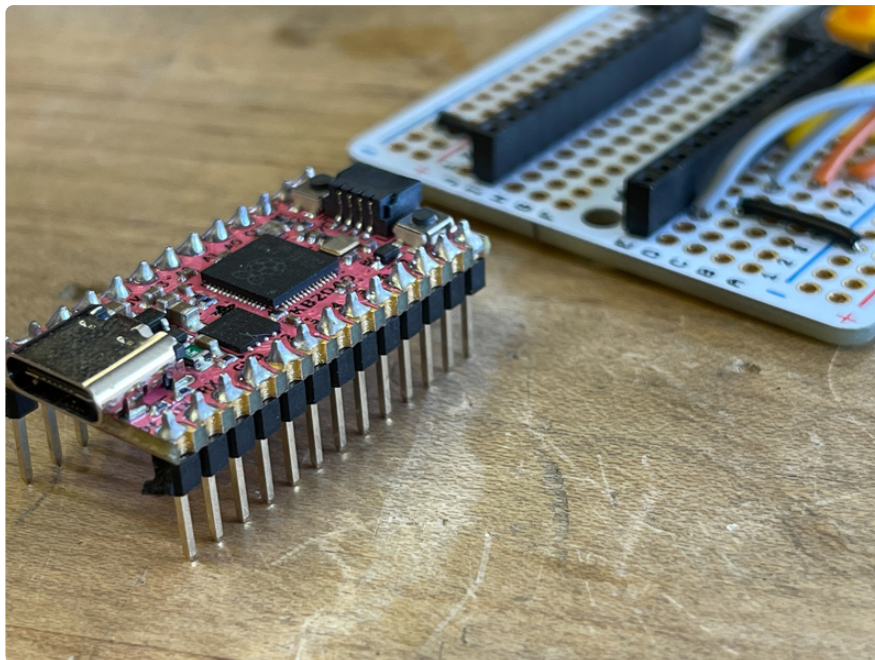


## Kee Boar Pins

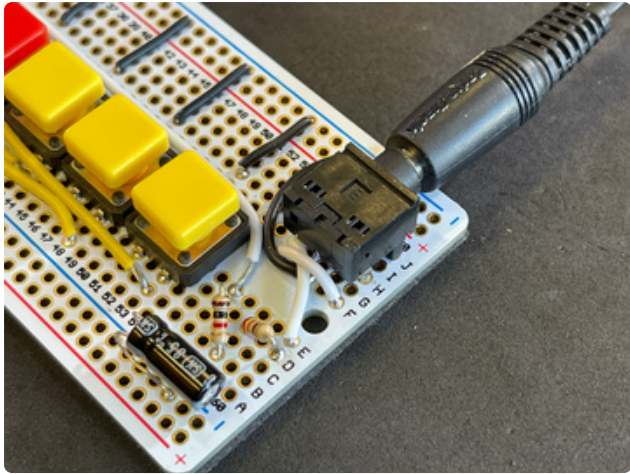
Solder header pins to the **Kee Boar** (similar to this **Feather** [configuration](https://adafru.it/EEm) (<https://adafru.it/EEm>)), then press it into the breadboard as shown in the diagram.



For the Perma Proto build you can solder the pins into the board or use optional header sockets as shown here.





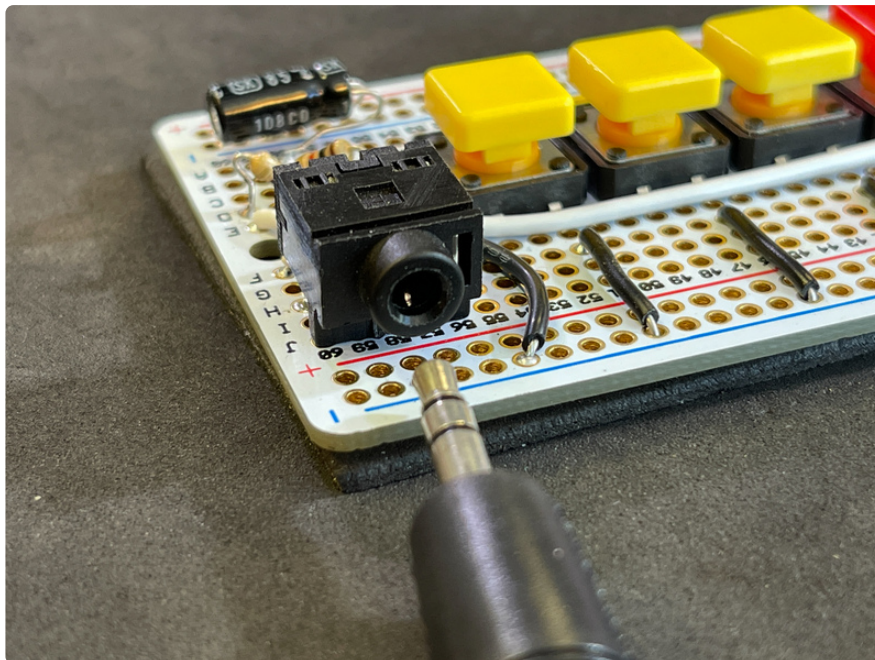
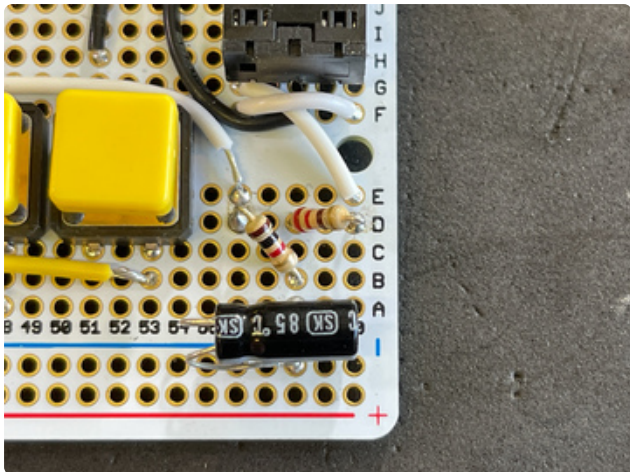


## Audio Noise Reduction

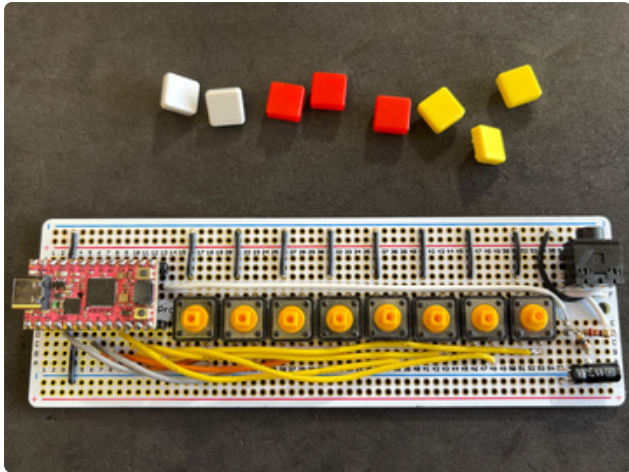
You'll build a small RC circuit to reduce some high end noise created by the PWM audio output of the board.

This consists of a  $\sim 1\text{k}\Omega$  resistor and  $\sim 2.2\mu\text{F}$  electrolytic capacitor that cut high frequency signal. There is also a  $220\Omega$  resistor to help reduce pops when plugging the audio cable in or out.

This runs the filtered signal to the 3.5mm audio jack. Since we're using mono audio, the left and right channels of the TRS jack are jumpered together.

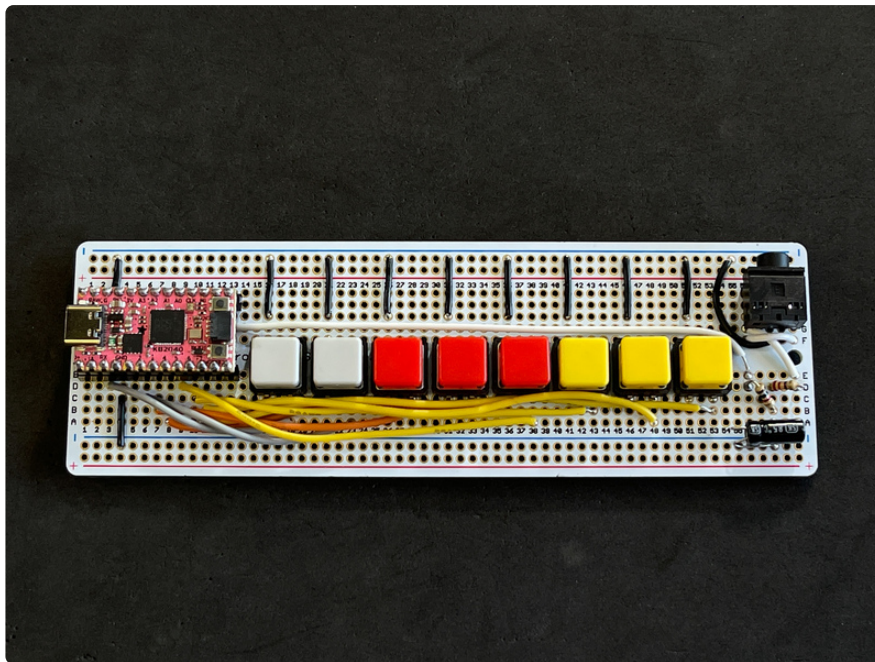


Note: Audio out on this project is LINE OUT level only, not AC-coupled for headphones, so is best used with powered speakers or an amplifier. It may drive small headphones, but not loudly.



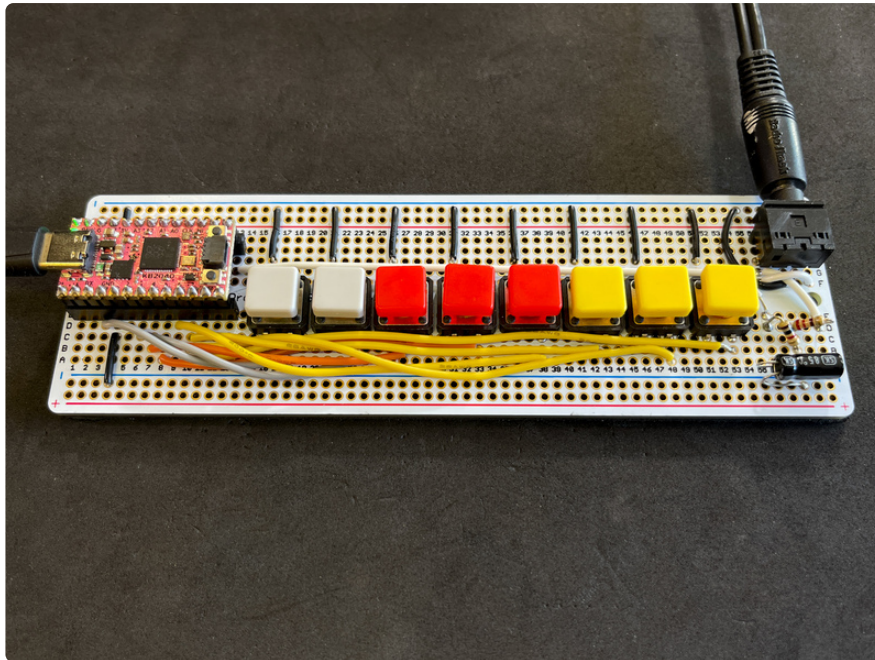
## Buttons

You'll add a set of eight tactile switches to trigger the different audio loops. These all run from individual GPIO digital input pins to ground on the KB2040.



Next, you plug in USB to power and program the Kee Boar, and then a 3.5mm stereo cable to an amplifier or powered speaker.





---

## Code the Breakbeat Breadboard

### Text Editor

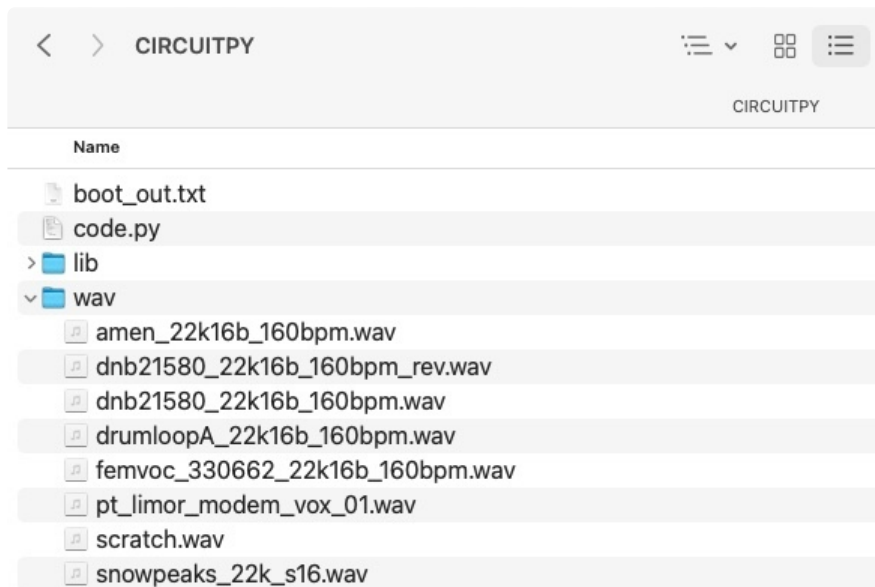
Adafruit recommends using the **Mu** editor for editing your CircuitPython code. You can get more info in [this guide \(https://adafru.it/ANO\)](https://adafru.it/ANO).

Alternatively, you can use any text editor that saves simple text files.

### Download the Project Bundle

Your project will use a specific set of CircuitPython libraries, folder of .wav file assets, and the **code.py** file. To get everything you need, click on the **Download Project Bundle** link below, and uncompress the .zip file.

Drag the contents of the uncompressed bundle directory onto your KeeBoard's **CIRCUITPY** drive, replacing any existing files or directories with the same names, and adding any new ones that are necessary.



```
# SPDX-FileCopyrightText: 2022 John Park for Adafruit Industries
# SPDX-License-Identifier: MIT

# Breakbeat Breadboard based on:
# @todbot / Tod Kurt - https://github.com/todbot/plinkykeeb
# Convert files to appropriate WAV format (mono, 22050 Hz, 16-bit signed) with
# command:
# sox loop.mp3 -b 16 -c 1 -r 22050 loop.wav
# put samples in "/wav" folder

import time
import board
import keypad
import audiocore
import audiomixer
from audiopwmio import PWMAudioOut as AudioOut

# wait a little bit so USB can stabilize and not glitch audio
time.sleep(3)

# list of (samples to play, mixer gain level)
wav_files = (
    ('wav/amen_22k16b_160bpm.wav', 1.0),
    ('wav/dnb21580_22k16b_160bpm.wav', 0.9),
    ('wav/dnb21580_22k16b_160bpm_rev.wav', 1.0),
    ('wav/drumloopA_22k16b_160bpm.wav', 1.0),
    ('wav/femvoc_330662_22k16b_160bpm.wav', 0.8),
    ('wav/scratch3667_4bar_22k16b_160bpm.wav', 0.5),
    ('wav/pt_limor_modem_vox_01.wav', 0.4),
    ('wav/snowpeaks_22k_s16.wav', 0.8),
    ('wav/dnb21580_22k16b_160bpm_rev.wav', 1.0)
)

# pins used by keyboard
KEY_PINS = (
    board.RX, board.D2, board.D3, board.D4,
    board.D5, board.D6, board.D7, board.D8
)

km = keypad.Keys( KEY_PINS, value_when_pressed=False, pull=True)

audio = AudioOut( board.D10 ) # RP2040 PWM, use RC filter on breadboard
mixer = audiomixer.Mixer(voice_count=len(wav_files), sample_rate=22050,
channel_count=1,
                        bits_per_sample=16, samples_signed=True)
audio.play(mixer) # attach mixer to audio playback
```



```

for i in range(len(wav_files)): # start all samples at once for use w handle_mixer
    wave = audiocore.WaveFile(open(wav_files[i][0], "rb"))
    mixer.voice[i].play(wave, loop=True)
    mixer.voice[i].level = 0

def handle_mixer(num, pressed):
    voice = mixer.voice[num] # get mixer voice
    if pressed:
        voice.level = wav_files[num][1] # play at level in wav_file list
    else: # released
        voice.level = 0 # mute it

while True:
    event = km.events.get()
    if event:
        if event.key_number < len(wav_files):
            if event.pressed:
                handle_mixer(event.key_number, True)

            if event.released:
                handle_mixer(event.key_number, False)

```

## Upload the Code and Libraries to the KB RP2040

After downloading the Project Bundle, plug your KB2040 into the computer USB port. You should see a new flash drive appear in the computer's File Explorer or Finder (depending on your operating system) called **CIRCUITPY**. Unzip the folder and copy the following items to the KB2040's **CIRCUITPY** drive.

- **lib** folder
- **wav** folder (this contains eight sample loop wav files)
- **code.py**

## Custom Sample Loops

To create your own loop wav files, convert your audio files to 16-bit mono WAV files at 22KHz sample rate. [This guide \(https://adafru.it/CQM\)](https://adafru.it/CQM) shows how.